COBB

INSIDE DOS®™

June 1993
Vol. 4 No. 6

Tips & techniques for MS-DOS & PC-DOS Versions 5 and 6

Organizing work groups in the DOS Shell

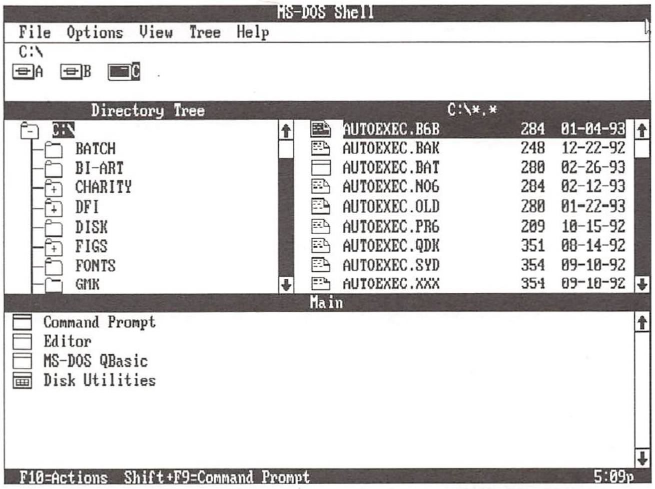
By Charity B. Edelen

Wouldn't you love to have everything you need for a project in one place—not just document files, but also their associated programs, any batch files, and even DOS commands you might need? Well, you can by setting up a program group for the project in the DOS Shell. In a program group, also known as a work group, you can gain easy access to every program, file, and DOS command you need for a project, even if the files are stored in different directories. In this article, we'll explain how program groups work and show you how to set one up and add the items you need for your project.

Understanding program groups

Think of a program group as a customized menu. When you select an item from a menu, you issue a command. Likewise, when you select an item from a program group, you tell DOS to switch to the directory where the item is stored and run the program. For example, suppose you've set up a Business Plan program group that includes Microsoft Word, Lotus 1-2-3, and dBASE, as well as related files you've created with these programs. When you select a program, you tell DOS to switch to that program's directory and load the file that executes the program. Thus, a program group doesn't store actual copies of programs or files—it just provides a route to the directories that house them.

Figure A


MS-DOS Shell
File Options View Tree Help
A
B
C
Directory Tree
Main
F10=Actions Shift+F9=Command Prompt
5:09p

```
C:\
C:\*.*
C:\
  BATCH          AUTOEXEC.B6B   284  01-04-93
  BI-ART         AUTOEXEC.BAK   248  12-22-92
  CHARITY        AUTOEXEC.BAT   280  02-26-93
  DFI            AUTOEXEC.N06   284  02-12-93
  DISK           AUTOEXEC.OLD   280  01-22-93
  FIGS           AUTOEXEC.PR6   209  10-15-92
  FONTS          AUTOEXEC.QDK   351  08-14-92
  GMK            AUTOEXEC.SYD   354  09-10-92
                 AUTOEXEC.XXX   354  09-10-92

Command Prompt
Editor
MS-DOS QBasic
Disk Utilities
```

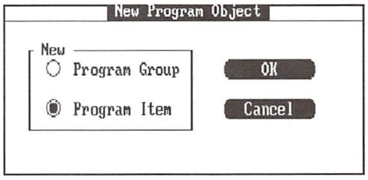
You must be in the Main window to set up a program group.

Setting up a program group

You set up program groups in the Main window shown at the bottom of Figure A. To display this window, you select Program/File Lists or Program List from the View menu. Then, to activate the Main window, you click in it or press [Tab] until the title bar appears highlighted.

With the Main window active, you issue the New... command from the File menu to begin setting up a new program group. This command opens the New Program Object dialog box, shown in Figure B. In this dialog box, you select the Program Group option, then click OK or press [Enter] to open the Add Group dialog box, shown in Figure C on the next page.

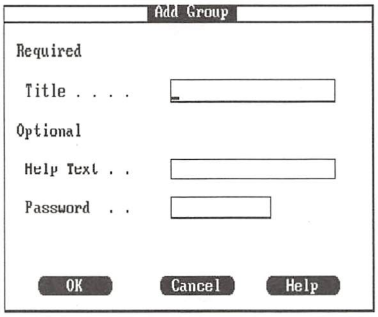
Figure B



The Program Group option in the New Program Object dialog box sets up a program group.

A Publication of The Cobb Group

Figure C

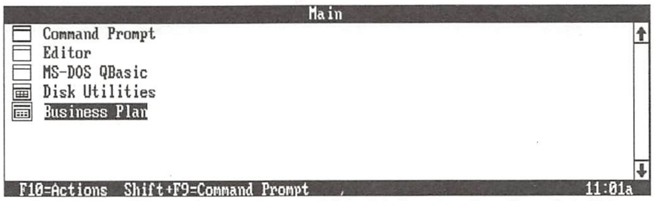


The Add Group dialog box lets you name the program group, enter text for the Help window, and assign a password to the group.

As Figure C indicates, the only item you have to enter in the Add Group dialog box is a title for the program group. The title you assign can contain up to 23 characters, including spaces. If you want, you can also enter information in the Help Text field. Anything you enter in this field will appear in the Shell Help window when you highlight the program group and press [F1]. Finally, you can assign a password to the program group to prevent unauthorized access to the project. (We'll explore the Help window and passwords in future issues of *Inside DOS*.)

After making entries in the Add Group dialog box, you click OK or press [Enter] to close it. When you do, your new program group will appear at the bottom of the list in the Main window. For instance, if you create a program group called *Business Plan*, it will appear at the bottom of the list in the Main window, as shown in Figure D. The ▤ icon indicates that this item is a program group.

Figure D



When you set up a program group, Shell adds it to the bottom of the Main window list, preceded by a program group icon.

Once you set up a new program group, Shell automatically adds one item—the Main group. It adds this item so that you can close the program group and return to the Main window. However, you have to manually add any other programs, files, and DOS commands that you might need. Next, let's look at how you add other items.

Adding items to a program group

Before you can add items to a program group, you must open the group. You can double-click the group or high-

INSIDE DOS™

Inside DOS (ISSN 1049-5320) is published monthly by The Cobb Group.

Prices

Domestic:	$49/yr. ($6.00 each)
Outside U.S.:	$69/yr. ($8.50 each)

Phone

Toll free:	(800) 223-8720
Local:	(502) 491-1900
Customer Relations Fax:	(502) 491-8050
Editorial Department Fax:	(502) 491-4200

Address

You may address tips, special requests, and other correspondence to

The Editor, *Inside DOS*
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

For subscriptions, fulfillment questions, and requests for bulk orders, address your letters to

Customer Relations
9420 Bunsen Parkway, Suite 300
Louisville, KY 40220

Postmaster

Second class postage is paid in Louisville, KY. Send address changes to

Inside DOS
P.O. Box 35160
Louisville, KY 40232

Back Issues

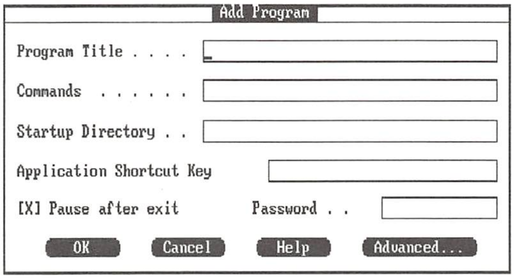
To order back issues, call Customer Relations at (800) 223-8720. Back issues cost $6.00 each, $8.50 outside the U.S. You can pay with MasterCard, VISA, Discover, or American Express, or we can bill you. Please identify the issue you want by the month and year it was published. Customer Relations can also provide you with an issue-by-issue listing of all the articles that have appeared in *Inside DOS.*

Staff

Editor-in-Chief:	Suzanne Thornberry
Assistant Editor:	Charity B. Edelen
Contributing Editor:	Van Wolverton
Editing:	Elizabeth Welch Timothy E. Hampton
Production Artist:	Julie Jefferson
Design:	Karl Feige
Publications Manager:	Tara Dickerson
Circulation Manager:	Brent Shean
Publications Director:	Linda Baughman
Editorial Director:	Jeff Yocom
Publishers:	Mark Crane Jon Pyles

light it with the ↓ or ↑ key and press [Enter] to open it. When you do, Shell replaces the Main window with one that lists the items in the group you opened and displays the group's title in the title bar. To add an item, you again issue the New… command from the File menu. This time, however, you leave Program Item selected in the New Program Object dialog box and click OK or press [Enter] to open the Add Program dialog box, shown in Figure E.

Figure E



Add Program

Program Title . . . .

Commands . . . . . .

Startup Directory . .

Application Shortcut Key

[X] Pause after exit

Password . .

In the Add Program dialog box, you give DOS all the information it needs to find the file, program, or command you want.

In the Program Title text box, you type the name you want to appear in the program window for that item. For example, if you're adding Microsoft Word to the group, you might type *MS Word* in this text box.

In the Commands text box, you type the command you'll use to run the item you're adding, including the path if it isn't specified in your AUTOEXEC.BAT file. For Microsoft Word, you might type *c:\word\word* in this text box, where *c:\word* is the Word program directory and the second *word* is the command that runs the program. If the item you're adding is a Word file called BPDRAFT.DOC stored in the \BPDOCS directory, you type

```
c:\word\word c:\bpdocs\bpdraft.doc
```

to open the BPDRAFT.DOC file in Word. If you're adding a batch file, you enter the path and name of the batch file you want to run, and if you're adding a DOS command, you simply enter the command. You can enter more than one command in the Commands text box, but you must separate them with a semicolon followed by a space.

The Program Title and Commands fields are the only two you have to fill in. But depending on your needs and preferences, you might want to use the other fields as well.

In the Startup Directory text box, you can specify a directory you want to switch to when the program starts. In the Application Shortcut Key text box, you can assign an [Alt], [Ctrl], or [Shift] key combination to the program by moving your cursor to this field and pressing the key combination. After starting the program from the DOS Shell, you can switch to it from any other active program by pressing this key combination. The Pause After Exit check box lets you see any error messages DOS might display when you exit the program. If you deselect this check box, the error messages will go by too quickly for you to read them. After displaying error messages, Pause After Exit asks you to press any key to return to the Shell and end the pause. Finally, in the Password text box, you can assign a password to the item if you want to limit access to it.

After assigning a title, specifying commands, and filling in any other appropriate fields, you click OK to add the item to the program group. Shell adds the item to the bottom of the program list, preceded by the 🗔 program icon.

Running the programs in a group

When you want to run a program in a program group, you simply open the group and then double-click the item or highlight it by using the ↓ or ↑ key and pressing [Enter]. Then, when you've finished using the program, you exit it and press any key to return to the Shell.

Instead of opening just one item in a group at a time, you might want several items open at once. If so, you'll need to turn on the Task Swapper before opening any items. The Enable Task Swapper command on the Options menu lets you switch back and forth among several programs without exiting them. (See "Program Switching with the DOS 5 Shell," which appeared in the October 1991 issue, for more on using the Task Swapper.) The beauty of setting up program groups is that once you open the group, opening subsequent items in the group is easy. You simply press [Ctrl][Esc] to return to the group window in the Shell and select the item you want to run. You don't have to switch groups or directories—you'll find everything you need right there in the program window.

A model work group

To get a feel for how program groups work, let's set up a model work group. Begin in the Main window of the DOS Shell and issue the New… command from the File menu. When the New Program Object dialog box appears, select Program Group and click OK or press [Enter]. Then, type *Business Plan* in the Title text box of the Add Group dialog box and again click OK or press [Enter]. Back in the Main window, you'll see that you've just added the Business Plan work group.

Now let's add a program, a file, and a DOS command to this group. First, double-click the Business Plan group in the Main window or highlight it by pressing the

↓ key and then press [Enter]. When the Business Plan window opens, you'll notice that the only item it contains is the Main group. Let's add a spreadsheet program to this group.

To start, again issue the New... command. This time, just click OK or press [Enter] to accept Program Item in the New Program Object dialog box. When the Add Program dialog box appears, type *Lotus 1-2-3* or the name of your spreadsheet program in the Program Title dialog box. Next, tab to the Commands text box and type *c:\123r23\123* or the path and command you need to run your spreadsheet. Then, tab to the Application Shortcut Key field and press [Ctrl]L. Finally, click OK or press [Enter]. In the Business Plan window, you'll see a new program item for your spreadsheet.

Next, let's add a file to the Business Plan group. Repeat the steps described in the previous paragraph, but this time type *October File* (or a descriptive name of any existing word processor document) in the Program Title text box. In the Commands text box, you type the line

```
c:\word\word c:\word\october.doc
```

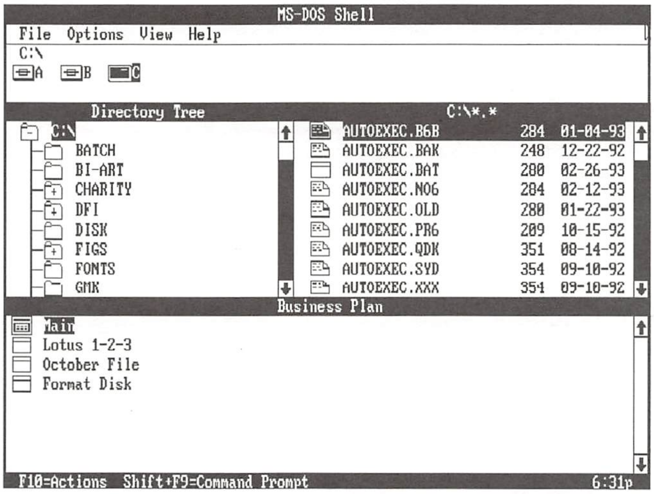
or the path and command that loads your word processor and opens the file you described in the Program Title field. Leave the Application Shortcut Key text box blank and click OK or press [Enter] to close the Add Program dialog box.

Finally, let's add the FORMAT command to this group. Once again, repeat the steps to open the Add Program dialog box. In the Program Title text box, you type *Format Disk* and in the Commands text box, type *format a:*, where *a* represents your floppy drive. Then, you simply click OK or press [Enter] to return to the Business Plan window. This window now looks like the one shown in Figure F.

To demonstrate the usefulness of this program group, let's open the items in it. But first, select Enable Task Swapper from the Options menu so you can switch between programs.

Now, double-click the spreadsheet item (Lotus 1-2-3) or highlight it and press [Enter] to launch your spreadsheet program. Then, open an existing file. Now, suppose you want to save this file to a floppy but you don't have a formatted diskette on hand. Since you set up an item in the Business Plan group that issues the FORMAT command, you can use that item to format a disk quickly. Press [Ctrl][Esc] to return to the Business Plan window in the Shell and then double-click the Format Disk item or highlight it and press [Enter]. When you do, DOS will execute the FORMAT command and ask you to insert a disk in your floppy drive. If you have a disk to format, insert it and press [Enter]; otherwise, just press [Enter] and then type *n* and press [Enter] to end the command. As soon as the command ends, DOS prompts you to press a key to return to the Shell.

Figure F



```
                              MS-DOS Shell
 File  Options  View  Help
 C:\
 A  B  C

        Directory Tree                         C:\*.*
 C:\                          AUTOEXEC.B6B        284  01-04-93
   BATCH                      AUTOEXEC.BAK        248  12-22-92
   BI-ART                     AUTOEXEC.BAT        280  02-26-93
   CHARITY                    AUTOEXEC.N06        284  02-12-93
   DFI                        AUTOEXEC.OLD        280  01-22-93
   DISK                       AUTOEXEC.PR6        209  10-15-92
   FIGS                       AUTOEXEC.QDK        351  08-14-92
   FONTS                      AUTOEXEC.SYD        354  09-10-92
   GMK                        AUTOEXEC.XXX        354  09-10-92
                           Business Plan
 Main
 Lotus 1-2-3
 October File
 Format Disk

 F10=Actions  Shift+F9=Command Prompt                            6:31p
```

A typical program group might contain a word processing program, some document files, and some DOS commands.

After you press a key, the Business Plan window reappears. Now, suppose you want to save the October File to the disk you just formatted. You haven't loaded your word processor yet, but you can load it and open the OCTOBER.DOC file at the same time. Just double-click the October File item or highlight it and press [Enter].

Once you save the file to the floppy disk, return to your spreadsheet program by pressing [Ctrl]L, the key combination you assigned when you added the program to the Business Plan group. Then, save the open file to the floppy disk if you want to and exit the program. When DOS prompts you, press a key to return to the Shell. Next, press [Ctrl][Esc] to switch to the file OCTOBER.DOC and exit your word processing program. Again, press a key to return to the Shell. Finally, double-click the Main group or highlight it and press [Enter] to close the Business Plan group and return to the Main window.

Conclusion

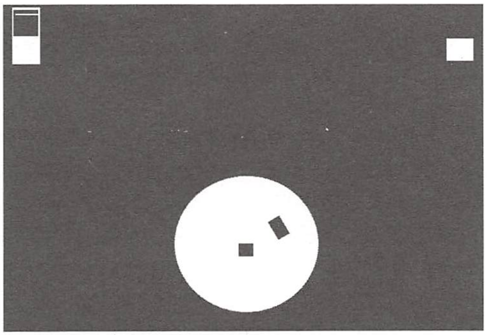
By setting up a program group every time you start a new project, you minimize problems, such as misplaced files, full directories, or wasted time, that tend to occur when your project is well under way. If you regularly undertake projects that combine files from different programs, work groups in the DOS Shell are a must.

SECURITY TIP

Write-protecting diskettes helps prevent unwanted changes

Have you ever wanted to ensure that no one accidentally changes the data on a diskette? For example, let's say you want to protect the final version of all the documents in a large project. If you've copied these files to a diskette, you can protect them very easily. As we'll see, you can physically alter both 3.5" and 5.25" diskettes to write-protect them. As the name implies, you can read data from a write-protected diskette, but you can't write changes to the diskette.

Figure A



Every 3.5" diskette includes a "switch" to let you write-protect the data on it.

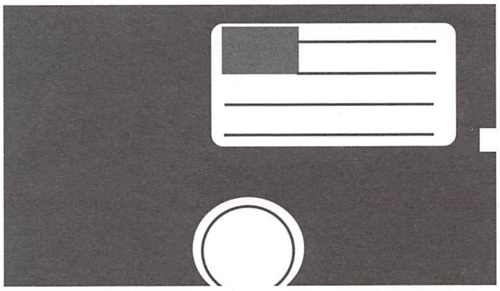
3.5" diskettes

Protecting a 3.5" diskette is especially easy. When you look at the back of the diskette, you'll see a small, rectangular indentation in the upper-left corner. A square tab slides back and forth within the rectangle. Normally, the tab, which is a switch for write-protecting the diskette, is at the bottom of the rectangle. To write-protect the diskette, you simply slide the tab to the upper half of the rectangle to expose a square opening, as shown in Figure A. It's as easy to unprotect the diskette—just slide the tab back down to its initial position. That's an advantage if you might later want to recycle the diskette, but it's a drawback, too: Anyone who knows this trick and wants to tamper with the data can overcome your precaution.

5.25" diskettes

The larger 5.25" diskettes don't have a built-in switch like the 3.5" diskettes. Instead, 5.25" diskettes have a notch on the right side of the diskette, as shown in Figure B. You write-protect the diskette by placing a self-stick tab over the notch. The box your diskettes came in contains a supply of these self-stick tabs. Be careful to place the tab on the diskette smoothly. You might face a repair bill if the tab catches on something inside the diskette drive.

Figure B



You can write-protect a 5.25" diskette by placing a tab over the notch, which appears on the right in this figure.

SECURITY TIPS

Watching out for virus symptoms

Programmers who develop anti-virus software are constantly working to outwit programmers who develop computer viruses. Fortunately, most anti-virus software does an excellent job of detecting the changes that viruses make to your system. Utilities such as Symantec's Norton AntiVirus, Central Point's Anti-Virus, and McAffee Associates' VShield are constantly being revised to detect and destroy the new viruses. In addition, those of you upgrading to DOS 6 will be able to use Microsoft Anti-Virus, a utility licensed from Central Point.

But what if you're a DOS 5 user who isn't ready to upgrade and who doesn't own third-party utilities? How can you tell if your system is acting odd because of a virus or because of a less exotic hardware or software problem? Even without add-ons, you'll be able to detect many common viruses—although you may need utilities to cure the virus. In this article, we'll briefly outline some of the ways viruses latch on to your system. Keeping an eye out for signs of viruses can help you deal with these dangerous programs before they damage your system.

Virus symptoms

If you aren't using an anti-virus utility, you might want to be on the lookout for these virus warning signs:

- A mysterious change in the file's date, especially of a program (EXE) or command (COM) file

- A small drop in the amount of total conventional memory that CHKDSK reports

Now, let's take a closer look at how you can check for these conditions.

Checking program files

First, you can check to see if any of your DOS program files have been changed by comparing the files listed on your DOS installation disks with the ones on your hard disk. You can print the directory of each diskette by inserting the diskette into the A: drive and then entering the command

```
C:\>dir a:\*.* > prn
```

After you've printed all the directories, you can compare the dates on the original commands with the dates on the files in your C:\DOS directory. The dates should be the same on the diskettes and hard disk, though the files on the diskettes will be smaller than the expanded files on your hard disk. Remember to check the copy of COMMAND.COM in your root directory as well. If you notice a discrepancy in the dates, you may have a virus and should use an anti-virus utility. A more radical solution is to reformat the hard disk, which destroys all the programs and data stored on the disk.

Checking conventional memory

Since many viruses stay resident in a portion of your system's memory, you can sometimes detect a virus by finding a reduction in your system's total conventional memory. You can check this type of memory by issuing the CHKDSK command. The last two lines of the CHKDSK report will show the amount of conventional memory you have and the amount of memory that's free. For example, CHKDSK may report

```
655360  total bytes memory
569616  bytes free
```

In this "healthy" system, the line *total bytes memory* indicates 640 Kb (that's 655,360 divided by 1,024). But, if CHKDSK shows less than 640 Kb, a virus may be using up a portion of your conventional memory. For example, the Stoned virus reduces memory by about 2 Kb, so an infected system might show 653,312 total bytes of memory. Again, if you suspect a virus, you'll probably want to check out your system with an anti-virus utility.

VAN WOLVERTON

Choosing a configuration at startup is easier with DOS 6

One of the ongoing frustrations of using DOS is trying to accommodate the rigid nature of the CONFIG.SYS file with the variations in how we use our systems. If a device that we use infrequently requires a large, memory-resident device driver, we don't want to sacrifice that memory all the time. In the past, the only solution was to create two versions of your CONFIG.SYS—one with the device driver and one without. To change configurations, you'd have to rename the current CONFIG.SYS file, name a new CONFIG.SYS, and then restart DOS with a different version of the CONFIG.SYS file. If we have several such devices—some of which might require commands in AUTOEXEC.BAT as well as CONFIG.SYS—we find ourselves with several versions of CONFIG.SYS and AUTOEXEC.BAT in the root directory. Previous issues of *Inside DOS* have described various ways of automating the reconfiguration process, but they all involve exchanging versions of the CONFIG.SYS and AUTOEXEC.BAT files and restarting the system.

DOS 6 eliminates the need for these hammer-and-tong techniques with an approach that's almost elegant. With just a few commands, you can set up a CONFIG.SYS file that displays a menu of your configuration choices each time you start or restart the system. (Note that you should always have a boot diskette on hand whenever you revise your CONFIG.SYS file.)

The basic technique is simple. You just divide your CONFIG.SYS into sections, or *blocks*. MENUITEM commands specify the names of the menu choices. A separate block for each menu item specifies the configuration commands DOS carries out if that menu item is chosen. Each block starts with a line that consists of the name of the block enclosed in brackets. You can put configuration commands to be carried out regardless of which menu item is chosen at the beginning of the CONFIG.SYS file (before the menu block), or you could put those commands in a separate block that's named COMMON. Figure A shows the general form you'd use to set up CONFIG.SYS for multiple configurations. We've used REM statements to note where you'd insert your own configuration commands.

Figure A

```
REM Commands common to all choices can go here [menu]
menuitem=choice1
menuitem=choice2

[choice1]

REM Choice1 commands go here

[choice2]
REM Choice2 commands go here

[common]
REM Commands common to all choices can go here
```

To define multiple configurations, you divide the CONFIG.SYS file into blocks, which are identified by names enclosed in brackets.

When you start your system with a CONFIG.SYS file set up like Figure A, the first thing you see is a menu that asks you to pick *choice1* or *choice2*. After you choose one of the items, DOS carries out the commands required for that configuration, carries out the commands common to all choices (if any) at the end, and then completes the startup process by carrying out the commands in AUTOEXEC.BAT.

The real world isn't so simple

As usually happens, though, the way we use our computer is more complex than the simple example shown in Figure A. Suppose your system is linked to a network and the terminate-and-stay-resident (TSR) network programs take up quite a bit of memory. You frequently use desktop publishing (DTP) and computer-assisted design (CAD) programs, which require device drivers and other TSR programs that also take up quite a bit of memory. You don't always have to be linked to the network, and you'd like to keep as much memory available as possible in your system by choosing one of four configurations when you start your system:

- *No network*, which doesn't start the network software but loads all the device drivers and other programs required by both DTP and CAD.
- *Network*, which starts the network but doesn't load any of the programs required for DTP and CAD.
- *Network with DTP*, which starts the network and loads the programs required for DTP.
- *Network with CAD*, which starts the network and loads the programs required for CAD.

Let's also suppose you want the system to default to *No network* if you don't respond within 30 seconds and you'd like the menu to be displayed in color. To change configurations, you'd simply press [Ctrl][Alt][Del] and choose the new configuration. Figure B shows the CONFIG.SYS commands that make all this happen.

Figure B

```
dos = high
umb device = c:\dos\himem.sys
device = c:\dos\emm386.exe noems
numlock = off

[menu]
menuitem = no_net,No network
menuitem = net,Network

submenu = net_plus,Network with DTP or CAD
menudefault = net,30
menucolor = 14,6

[no_net]
devicehigh = c:\scangal\sjdriver.sys
devicehigh = c:\cad\hpplot.sys
devicehigh = c:\dos\ramdrive.sys 1024 512 128 /e

[net]
set net=yes
lastdrive = h

[net_plus]
menuitem = net_&_DTP,Network with DTP
menuitem = net_&_CAD,Network with CAD

[net_&_DTP]
include = net
devicehigh = c:\scangal\sjdriver.sys

[net_&_CAD]
include = net
devicehigh = c:\cad\hpplot.sys
devicehigh = c:\dos\ramdrive.sys 1024 512 128 /e

[common]
devicehigh = c:\dos\dblspace.sys /move
devicehigh = c:\dos\ansi.sys
devicehigh = c:\dos\mouse.sys
buffers = 20,0
files = 30
fcbs = 1,0
stacks = 0,0
shell = c:\dos\command.com c:\dos\ /e:1000 /p
```

The sample CONFIG.SYS file lets users choose from various network and specialized standalone configurations.

Although our sample four-choice CONFIG.SYS file has several blocks, the structure is still fairly simple. The CONFIG.SYS file always carries out the first four commands. The first three are the usual ones that set up memory management on a 386- or 486-based PC. The fourth command is a newcomer, NUMLOCK, that lets you control the status of the Num Lock key at startup. If you prefer to leave Num Lock on and not use the numeric keypad as cursor control keys, you wouldn't include this command.

The menu block

Next comes a block of five commands headed by the line [menu]. This is the *menu block*, which specifies the choices and behavior of the menu. Next, two MENUITEM commands specify two menu choices:

```
menuitem = no_net,No network
menuitem = net,Network
```

Let's look more closely at the first of these commands to see how MENUITEM commands work. The first part of this MENUITEM command specifies the block name for the menu choice, in this case, *no_net*. Later, you'll use this block name as the heading for a block of commands. Since block names can't contain spaces, you'll probably want to specify menu text in the MENUITEM command. To include special menu text, type a comma after the block name, then type the text exactly as you want it to appear on the menu. For the *no_net* configuration, our menu will display *No network*.

The next command is

```
submenu = net_plus,Network with DTP or CAD
```

The SUBMENU command specifies a menu choice that results in the display of another menu, allowing you to nest menus for more detailed choices. In this case, selecting *Network with DTP or CAD* will present the submenu detailed in the [net_plus] block.

The fourth command

```
menudefault = net,30
```

tells DOS to select the first item (the one named *net*) if there's no response within 30 seconds of displaying the menu.

The last command in the menu block

```
menucolor = 14,6
```

displays the startup menu in bright yellow characters on a brown background. If you don't care about displaying the menu in color, you can omit the MENUCOLOR command. Table A shows you other codes you can specify to choose menu colors.

Should you install DoubleSpace?

In the accompanying article, "Choosing a Configuration at Startup Is Easier with DOS 6," Van Wolverton shows the following command in the [common] section of the sample CONFIG.SYS file:

```
devicehigh = c:\dos\dblspace.sys /move
```

This command installs the DoubleSpace driver (DBLSPACE.SYS), one of the new features in DOS 6. If you're upgrading, you'll need to decide if you want to use the DoubleSpace utility to roughly double the space available on your hard disk drive. Although DoubleSpace can be a boon if you're running out of space on your hard disk, installing the DoubleSpace driver is a commitment of sorts. Before you make the commitment to use DoubleSpace, you should consider two points.

First, DoubleSpace makes it impossible for you to return to your old version of DOS by using the UNINSTALL diskette that the DOS 6 Setup program creates. But chances are you'd never need to use the UNINSTALL diskette anyway.

Second, if you find some incompatibility or inconvenience in using DoubleSpace, uninstalling the compressed disk can take some time and patience. You'll need to back up all the data that's on the compressed drive, delete the compressed volume, and then restore the data. (You can find additional information on uninstalling DoubleSpace in the DOS 6 README.TXT file.)

Aside from the potential problems with removing DoubleSpace, the utility seems to be well integrated with DOS 6 and should be trouble-free for most users. And, if you do run into a problem with a compressed file, third-party vendors will be offering upgrades that work with DOS 6. In fact, at press time, Symantec had announced that Norton Utilities 7.0 would be able to recover data from files compressed with DoubleSpace.

Table A

Black	0	Gray	8
Blue	1	Bright blue	9
Green	2	Bright green	10
Cyan	3	Bright cyan	11
Red	4	Bright red	12
Magenta	5	Bright magenta	13
Brown	6	Yellow	14
White	7	Bright white	15

You can enter these codes with the MENUCOLOR command to choose colors for your configuration menus.

Now, let's look at how DOS uses the menu block commands when you boot your PC. When DOS starts to carry out the commands in this CONFIG.SYS file, it displays the menu shown in Figure C.

Figure C

```
MS-DOS 6 Startup Menu
MMMMMMMMMMMMMMMMMMMMM

1. No network
2. Network
3. Network with DTP or CAD

Enter a choice: 2 Time remaining: 30
```

The CONFIG.SYS file shown in Figure B will display this startup menu when a user boots up.

Because the MENUDEFAULT command specifies choice 2 (network) as the default, choice 2 is highlighted. The MENUDEFAULT command also specifies a time limit of 30 seconds, so the time remaining before DOS selects the default is counted down at the right side of the bottom line of the menu. If you don't specify a time limit, DOS waits forever and doesn't display a countdown timer; if you don't include a MENUDEFAULT command at all, choice 1 is highlighted.

Processing your choices

After displaying the configuration menu shown in Figure C, CONFIG.SYS will carry out the commands in the block you specify. The next two blocks in CONFIG.SYS, named [no_net] and [net], specify the commands to be carried out if you choose one of the first two menu items. If you choose *No network* by pressing 1, for example, the three DEVICEHIGH commands are carried out to load device drivers for a scanner and plotter and to set up a RAM drive. If you choose *Network* or allow DOS to select it by default, DOS creates an environment variable named NET and sets its value to *yes*. (The sidebar "Linking AUTOEXEC.BAT Commands to the Configuration You Choose" explains how you can use this NET variable.) The last command in the [net] block sets the highest allowable drive letter to H, which allows for network drives.

The next three blocks, which are named [net_plus], [net_&_DTP], and [net_&_CAD], specify the contents of the submenu named [net_plus], which we defined in the SUBMENU command in the [menu] block:

```
[net_plus]
menuitem = net_&_DTP,Network with DTP
menuitem = net_&_CAD,Network with CAD

[net_&_DTP]
include = net
devicehigh = c:\scangal\sjdriver.sys

[net_&_CAD]
include = net
devicehigh = c:\cad\hpplot.sys
devicehigh = c:\dos\ramdrive.sys 1024 512 128 /e
```

Because of the commands in this section, CONFIG.SYS will display the menu shown in Figure D if you choose the third item—*Network with DTP or CAD*—from the menu shown in Figure C.

Figure D

```
MS-DOS 6 Startup Menu
MMMMMMMMMMMMMMMMMMMMM

1. Network with DTP
2. Network with CAD

Enter a choice: 1
```

If you press 3 when DOS displays the menu shown in Figure C, this submenu will present you with two more choices.

Like any submenu, the one shown in Figure D displays just the choices defined in the submenu block. In this case, the two MENUITEM commands in the [net_plus] block define the submenu choices. Then, the [net_&_DTP] and [net_&_CAD] blocks specify the commands to be carried out if the corresponding items are chosen from the submenu. Notice that each of these blocks begins with the command *include = net*. These INCLUDE commands tell DOS to carry out all the commands in the block named [net] just as if *Network* had been chosen.

The final block of the CONFIG.SYS file, named [common], specifies the commands to be carried out before leaving CONFIG.SYS, regardless of which menu item was chosen.

A note on MEMMAKER

You might be wondering how the multiple configuration technique works with the new MEMMAKER utility, which automates the formerly tedious task of maximizing available memory on 386- and 486-based computers. (We discuss the MEMMAKER utility briefly in the letter "MEMMAKER Can Free Up Conventional Memory," which appears on page 11.) You might find MEMMAKER doesn't manage your memory as well when your CONFIG.SYS file contains either INCLUDE commands or [common] blocks, as this example does. Microsoft's DOS 6 *User's Guide* suggests that you create separate CONFIG.SYS files and run MEMMAKER for each of the configurations, then combine the optimized files into your menued CONFIG.SYS file. But, if you use DEVICEHIGH instead of DEVICE to load all device drivers and put LOADHIGH (or LH) before each DOS command in an AUTOEXEC.BAT file that runs a TSR program, you should do almost as well.

The ability to create multiple configurations is the sort of feature that doesn't grab a lot of headlines but enhances the usability of your system every time you use it. Take the time to set up CONFIG.SYS, and you'll thank yourself every time you start your computer.

Contributing editor Van Wolverton is the author of the best-selling books Running MS-DOS *and* Supercharging MS-DOS. *Van, who has worked for IBM and Intel, currently lives in Alberton, Montana.*

Linking AUTOEXEC.BAT commands to the configuration you choose

By Van Wolverton

If some of your configuration options require commands in the AUTOEXEC.BAT file, DOS offers a couple of ways to carry out commands specific to a startup menu choice. Remember that if you choose the *Network* configuration when you use the sample configuration file on page 7, DOS creates an environment variable named NET with a value of *yes*. If the network configuration requires a DOS command rather than a configuration command, you can carry out that command in the AUTOEXEC.BAT file by testing the value of the environment variable NET.

For example, suppose you use a batch file named STARTNET.BAT in the C:\LANTASTI directory to start your network programs. If so, you need only add the following command to AUTOEXEC.BAT:

```
if "%net%"=="yes" call c:\lantasti\startnet
```

Checking for the NET variable is fine if you need to test for only one configuration in AUTOEXEC.BAT. But if *several* of your configuration choices require DOS commands, you can take advantage of a system-generated environment variable to carry out the required commands. When you choose an item from the startup menu, DOS creates an environment variable named CONFIG and sets its value to the block name of the menu item (not the optional menu text you assigned the item).

You can use the CONFIG variable to tell DOS which portion of the AUTOEXEC.BAT file you want to carry out. Simply type a label for each block name used in your CONFIG.SYS file. Underneath each label, enter the AUTOEXEC.BAT commands you want to correspond to that configuration. End each of these sections with a GOTO END command. Then, place an :END label at the end of your AUTOEXEC.BAT file. You can carry out the required commands by including the following GOTO command in AUTOEXEC.BAT:

```
goto %config%
```

The AUTOEXEC.BAT file will jump to the label specified in the CONFIG environment variable, execute those commands, and then go to the :END label to quit. For example, if you chose *No network* from your configuration menu, the AUTOEXEC.BAT shown in Figure A would carry out the commands under the :NO_NET label, then quit.

Figure A

```
@echo off
rem You can put any commands you always need here.
goto %config%

:no_net
rem The commands needed when you aren't on the network
rem go here.
goto :END

:net_&_DTP
rem The commands needed with desktop publishing go
rem here.
goto :END

:net_&_CAD
rem The commands needed with the CAD program go here.
:END
```

You can segment your AUTOEXEC.BAT file into the same blocks as your CONFIG.SYS menu blocks.

Extra TODAY.BAT files report the wrong time

I created the TODAY.BAT file as described in the article "Capturing the Date and Time as Environment Variables," which appeared in the April 1993 issue of *Inside DOS*. I've also set up my AUTOEXEC.BAT file to report the date and time as you described. However, when I rebooted my computer several times to test the file, I noticed that the time stayed the same. Is there a problem with TODAY.BAT?

Paul McKnight
Bellevue, Washington

We're sorry to report a problem with TODAY.BAT. The batch file works fine if you always run it from the C:\BATCH directory (or whichever directory you've chosen for your batch files). However, if you run it from another directory, DOS creates a second TODAY.BAT file in that directory. DOS creates the second file because the COPY command that updates the date and time stamps of the TODAY.BAT file will also copy the file into the current directory. The first time you run TODAY.BAT from a directory other than C:\BATCH, you'll still get the correct time. But the second time you run the batch file from that directory, the COPY command will update the time stamp of the TODAY.BAT file in the current directory, instead of the TODAY.BAT file in the C:\BATCH directory. Then the command

```
dir c:\batch\today.bat ¦ find "TODAY" > tempdate.bat
```

sends the directory listing of the C:\BATCH\TODAY.BAT file to the TEMPDATE.BAT file. So TEMPDATE.BAT gets the old date and time information from the TODAY.BAT file in the C:\BATCH directory.

The easiest way to fix the batch file is to add the command *cd \batch* before the COPY command. Figure A shows the new version of the batch file, with the new command printed in red.

Figure A

```
@echo off
rem TODAY.BAT creates DATE and TIME environment variables.
if not "%4"=="" goto :SETDATE
cd \batch
copy /b c:\batch\today.bat+ > nul
dir c:\batch\today.bat ¦ find "TODAY" > tempdate.bat
tempdate
:SETDATE
set date=%3
set time=%4
del tempdate.bat
```

This version of TODAY.BAT will work from any directory on your C: drive.

Adding *cd \batch* ensures that the COPY command updates the time stamp of C:\BATCH\TODAY.BAT only and prevents the COPY command from creating multiple copies of TODAY.BAT. (You can delete any copies of TODAY.BAT not in your batch file directory.) We apologize for any inconvenience the old version of TODAY.BAT may have caused you.

MEMMAKER can free up conventional memory

After installing DOS 6, I decided to run MEMMAKER to try to optimize my system's memory. I was pleased to discover that it worked very well. In fact, MEMMAKER freed an additional 150 Kb of conventional memory. With the new configuration, I can run my fax board while I'm connected to the network—a task that was impossible with my old configuration. Based on my experience, I'd encourage others who decide to upgrade to try MEMMAKER on their systems.

Joe Robison
Kansas City, Missouri

Thanks for sharing your experience. Although your results may not be as dramatic as Mr. Robison's, running MEMMAKER may help you increase the amount of conventional memory available on your system—if it's based on a 386 or later microprocessor.

You'll find MEMMAKER easy to use if you have a single CONFIG.SYS file and you want to improve it. (As Van Wolverton noted in the article "Choosing a Configuration at Startup Is Easier with DOS 6," which begins on page 6, using MEMMAKER with multiple configurations can be a bit more complicated.) The utility can also help you better manage memory with Microsoft Windows; in fact, MEMMAKER will edit your Windows SYSTEM.INI file if necessary. However, you shouldn't run MEMMAKER while Windows is running.

If you've upgraded to DOS 6, you can use the MEMMAKER utility to reconfigure your system by typing *memmaker* at the DOS prompt and pressing [Enter]. When you run MEMMAKER, it will ask you if you want to use the Express or Custom setup. If you feel you have a lot of expertise with configuration, you'll probably want to choose Custom. Otherwise, we suggest you use the Express setup. (We'll assume you're using Express in this brief look at MEMMAKER.)

MEMMAKER will ask if you're using any programs that require expanded memory, and you can answer Yes or No. If you aren't sure, you can accept the default of Yes. Then, MEMMAKER will check to see if you've installed Windows on your hard disk. Finally, the utility will tell you to remove any diskettes from your PC and press [Enter] to reboot.

As the computer reboots, MEMMAKER will examine your system and tweak your CONFIG.SYS file and possibly your AUTOEXEC.BAT and SYSTEM.INI files. Then it will pause and ask you to press [Enter] to reboot again. As the computer boots, watch for any error messages. Then, MEMMAKER will ask you if everything seems to be working properly. If you saw error messages, you can answer No by pressing N; otherwise, accept the default of Yes.

One final note: If your PC doesn't boot properly after using MEMMAKER, you can press [F5] when you see the message *Starting MS-DOS* in order to bypass the CONFIG.SYS and AUTOEXEC.BAT files when you boot up. Then, you can enter the command

```
C:\>memmaker /undo
```

to return to the previous version of your CONFIG.SYS and AUTOEXEC.BAT files.

DOS requires two DIR commands to check for hidden and system files

While I prepared to use the disk-copying technique presented in the letter "RAM Disks Can Speed Copying Files," which appeared in the March 1993 issue of *Inside DOS*, I discovered a problem in checking for hidden and system files. Your reply to the letter suggested using the command

```
dir b:\ /a:h /a:s /s
```

to check the source diskette for files having the hidden or system attributes. However, I found this form of the DIR command really only checks for the system attribute. In fact, it appears the DIR command will list only files that have the second of the attributes requested. For example, when I changed the order of the switches to */a:s /a:h*, the DIR command displayed only the hidden files.

As a workaround, I created a batch file named BTEST.BAT to check for hidden and system files on a diskette in the B: drive. The batch file I created contains only three commands:

```
@echo off
dir b:\ /a:h /s
dir b:\ /a:s /s
```

Placing the /A:H and /A:S switches in separate commands ensures that the DIR command reports *both* hidden and system files. Including the /S switch in each DIR command tells the batch file to search for hidden and system files contained in any subdirectories on the diskette.

If BTEST.BAT shows that the diskette has no hidden or system files, I use the XCOPY technique described in the March issue to copy the files. If the diskette has hidden or system files, I know I need to use the DISKCOPY utility.

Walter Bunch
Lake Arrowhead, California

Thanks for pointing out the problem with combining the two switches, and thanks for sharing your workaround with us. ■

DOS 6 lets you select CONFIG.SYS commands as your PC boots up

One new convenience included in DOS 6 is the ability to process the CONFIG.SYS file line by line. You can do this by pressing [F8] while the message *Starting MS-DOS...* appears during bootup.

After you press [F8], DOS will display the first line of the CONFIG.SYS file, followed by *[Y,N]*. You simply press Y to carry out the command or N to skip the command. For example, if your first command loads the HIMEM.SYS driver and you need to use this driver, DOS will display

```
DEVICE=C:\DOS\HIMEM.SYS [Y,N]?
```

Then, you can press Y to load the HIMEM.SYS extended memory manager.

Printed in the USA
Printed on recycled and recyclable paper
50% recycled fiber including 10% post-consumer waste paper